

---

Subject: Re: Unsupported keyword on older IDL version  
Posted by [Paul Hick](#) on Tue, 30 Jul 2002 03:30:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

William Thompson wrote:

```
> Paul Hick <pphick@ucsd.edu> writes:
>
>
>> This kind of problem must have come up in this group before, but here it
>> is anyways.
>>
>
>> If I call the IDL systime function with the utc keyword:
>> print, systime(/utc)
>>
>
>> on IDL versions older than 5.4 I get the expected 'keyword not
>> supported' error message, since, well, it's not supported, right? The
>> error message pops up when the function containing the systime call is
>> compiled, so testing e.g. !version.release before executing the systime
>> call doesn't work.
>>
>
>> The only solution I have come up with so far is to put the systime call
>> inside a call_function call:
>> print, call_function( 'systime', /utc)
>> This will compile (I will still need to test for !version.release before
>> actually executing the statement of course).
>>
>
>> This to me looks like a somewhat illegitimate use of call_function. Not
>> that I mind bending the rules a bit, but are there any other (better)
>> techniques of dealing with keywords that don't work in all IDL versions.
>>
>
>
> Why do you call this "illegitimate"? To me this seems like just the sort of
> thing that call_function() was designed for. I've always done it this way in
> the past. In fact, I've usually used execute() in the past, but
> call_function() is more elegant.
>
> Bill Thompson
>
```

My main use for call\_function has been as the IDL version of the Fortran 'external' mechanism: passing function names as arguments to procedures that would then call the function using call\_function. That's

a lot more powerful than the current keyword problem. But of course you are right: whatever works ....

Paul

---